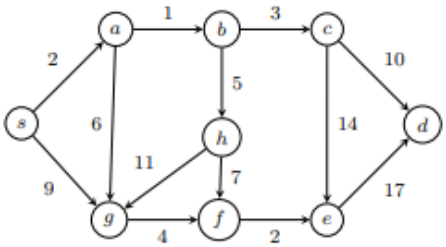
Exercise 1:

Use Dijkstra's algorithm to compute all shortest paths starting at node s. Show the values of the vertices after each iteration.



Exercise *2:*

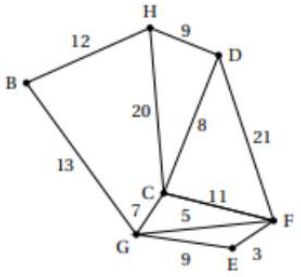
A weighted directed graph **G** is given by the incidence matrix below.

	s	a	b	c	d	t
s		19	8			
a			14		6	
b				4	22	
c		2			10	11
d						2
t						

1. Draw the graph G.
2. According to Dijsktra, what is the shortest path from **S** to **T**?

Exercise3*:*

Tourists are staying at a hotel H. A guide plans to take them on a tour of the region using the roads designated as points of tourist interest by the tourist office. The road sections he wants to take are shown on the graph below. The weight of each edge represents the distance in kilometers of the different road sections.



1. Can the guide take all the sections of road by passing each one once and only once, starting from the hotel and returning to it? Justify your answer.

2. Can the guide take all the sections of road by passing each one once and only once, starting from the hotel but not necessarily returning to it? Justify your answer.
3. A museum is located at **E**. Determine the shortest route from hotel **H** to museum **E**. Justify your answer.

Exercise4*:*

The table below gives the weights (lengths) w(x, y) of the arcs in a directed graph (x corresponds to the row, and y to the column).

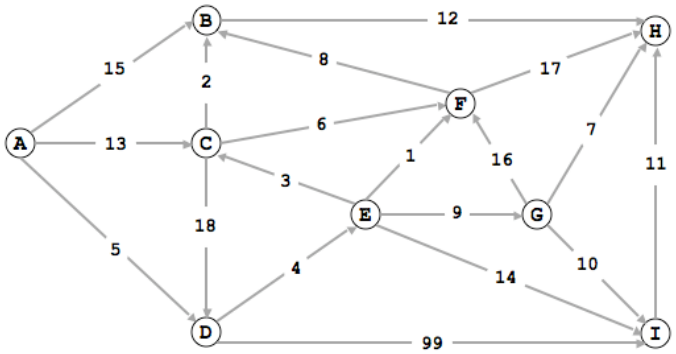
	A	B	C	D	E	F
A	0	8	8	8	8	2
B	8	0	−2	7	6	2
C	8	4	0	3	2	3
D	8	4	0	0	−1	4
E	8	2	2	3	0	8
F	8	3	2	−1	4	0

1. Use the method of Bellman-Ford to determine a shortest path from A to C, or a negative circuit if such a path does not exist.

2. Same questions, but now with w(C, D) = 0.

Exercise5*:*

Consider the following weighted directed graph.



1. Run Dijkstra's single source shortest path algorithm on the above digraph using C as the source. Give the order in which the vertices are visited. Note that A is not reachable from C, so it is never visited.

2. Give the distance of the shortest path from C to each vertex v (except A) and the last edge on the shortest path to v.

v	distTo[v]	edgeTo[v]
A	--	--
B		
C	0	null
D		
E		
F		
G		
H		
I		